Blue Team: Summary of Operations

Table of Contents

Network Topology

The following machines were identified on the network:

- Kali
  - **Operating System**: Linux
  - **Purpose**: Standard VM to attack other machines.
  - **IP Address**: 192.168.1.90
- Capstone
  - **Operating System**: Linux
  - **Purpose**: Target VM that tests filebeat, packetbeat, metricbeat logs to ELK server (Kibana).
  - **IP Address**: 192.168.1.105
- ELK
  - **Operating System**: Linux
  - **Purpose**: ELK server that hosts the Kibana dashboard that the logs and alerts are sent to.
  - **IP Address**: 192.168.1.100
- Target 1
  - **Operating System**: Linux
  - **Purpose**: Vulnerable WordPress Server. Logs sent to ELK server.
  - **IP Address**: 192.168.1.110
- Target 2
  - **Operating System**:
  - **Purpose**: Difficult WordPress Server. Logs sent to ELK server.
  - **IP Address**: 192.168.1.115

Description of Targets

The target of this attack was: Target 1 - 192.168.1.110

Target 1 is an Apache web server and has SSH enabled, so ports 80 and 22 are possible ports of entry for attackers. As such, the following alerts have been implemented:

Monitoring the Targets

Traffic to these services should be carefully monitored. To this end, we have implemented the alerts below:

Excessive HTTP Errors

Excessive HTTP Errors is implemented as follows:

- **Metric**: WHEN count() GROUPED OVER top 5 'http.response.status_code'
- **Threshold**: 400
- **Vulnerability Mitigated**: User Enumeration & Brute Force
- **Reliability**: High reliability. The threshold for error codes are 400+ and codes above 400 are usually client and server error responses thus giving more incentive for concern if the errors go off in high intervals.

HTTP Request Size Monitor

HTTP Request

Size Monitor is implemented as follows:

- **Metric**: WHEN sum() of http.request.bytes OVER all documents
- **Threshold**: 3500
- **Vulnerability Mitigated**: Code Injection (SQL, XSS, Command Inj) & DDoS
- **Reliability**: The alert may come with false positives due to its medium reliability. There is a possibility for very large HTTP traffic to be requested on top of legitimate HTTP traffic.

CPU Usage Monitor

CPU Usage Monitor is implemented as follows:

- **Metric**: WHEN max() OF system.process.cpu.total.pct OVER all documents
- **Threshold**: 0.5
- **Vulnerability Mitigated**: Malicious programs/software using CPU resources
- **Reliability**: With the threshold set at 0.5, it is highly reliable to determine running background programs that will aid in the improvement of CPU usage in order to differentiate normal programs from malicious ones.